

功能：计算定义域同为 $[0, 1] \times [0, 1]$ 的B样条曲面 $q(u, v)$ 对B样条曲面 $p(u, v)$ 的最大相对偏差 $\max d$ 及其在两曲面上对应的最大相对偏差点 $(\max qx, \max qy, \max qz)$ 与 $(\max px, \max py, \max pz)$ 。

输入参数：B样条曲面 $p(u, v)$ 由控制顶点为 $(m_xVertex, m_yVertex, m_zVertex)$ 与节点矢量 m_uNode 、 m_vNode 定义，B样条曲面 $q(u, v)$ 由控制顶点为 $(m_xRVertex, m_yRVertex, m_zRVertex)$ 与节点矢量 $m_uNewNode$ 、 $m_vNewNode$ 定义，均属受保护成员。

输出参数： $\max d$ —B样条曲面 $q(u, v)$ 对 $p(u, v)$ 间的最大相对偏差， $(\max qx, \max qy, \max qz)$ 、 $(\max px, \max py, \max pz)$ —在两曲面上对应的最大相对偏差点，属公共成员。

调用函数：GetQSurDerivat—求B样条曲面 $q(u, v)$ 上的点与偏导矢；QtoPSurDistance—求B样条曲面 $q(u, v)$ 上的点到B样条曲面 $p(u, v)$ 的距离，见15.1节。

```
void SurQtoPHausdorff()
{
    CPoint p0, p1;
    double a=0., b=1., c=0., d=1.;
    double u0=a, u1=b, v0=c, v1=d;
    double xq=0., yq=0.;
    double eps=0.0001;
    double uq, vq, up0, up1, vp0, vp1;
    double qx, qy, qz, px, py, pz, d0, d1, d_1, d00, d01, d10, d11;
    double ratio=(sqrt(5.)-1.)/2.; /* 黄金分割数0.618...的精确值 */
    double uh=(b-a)/10., vh=(d-c)/10.; /* 将定义域[a, b]x[c, d]等分成10x10子矩形
                                         域进行最大距离初搜索 */

    double sum=0;
    int imax, jmax;
    double aa, cc, u_1, v_1;
    /* 下面两个for循环进行最大偏差初搜索，确定11x11点中的最大距离点
    (imax, jmax), q(u, v)对于p(u, v)的最大距离点应在围绕该点的1^4个子矩
    形域内，通过比较确定一个子矩形域以左下角点序号(imax, jmax)标识。*/
    for(int i=0; i<10; i++)
    {
        aa=a+i*uh;
        for(int j=0; j<=10; j++)
        {
            cc=c+j*vh;
            u0=aa; v0=cc;
            GetQSurDerivat(0, 0, u0, v0, qx, qy, qz);
            uq=u0; vq=v0;
            QtoPSurDistance(uq, vq, qx, qy, qz, px, py, pz, d0);
            if(d0->maxd>1.E-10) {maxd=d0; imax=i; jmax=j; maxqx=qx;
                                maxqy=qy; maxqz=qz; maxpx=px; maxpy=py; maxpz=pz;}
        }
    }
    if(maxd<sqrt(2)) return;
    if(imax==0&&jmax==10) jmax=9;
    if(imax==10&&jmax==0) imax=9;
    if(imax==10&&jmax==10) {imax=9; jmax=9;}
    if(imax>0&&imax<10&&jmax>0&&jmax<10)
    {
        u_1=a+double(imax-1)*uh; v0=c+double(jmax)*vh;
        GetQSurDerivat(0, 0, u_1, v0, qx, qy, qz);
        uq=u_1; vq=v0;
        QtoPSurDistance(uq, vq, qx, qy, qz, px, py, pz, d_1);
        u1=a+double(imax+1)*uh; v0=c+double(jmax)*vh;
        GetQSurDerivat(0, 0, u1, v0, qx, qy, qz);
        uq=u1; vq=v0;
        QtoPSurDistance(uq, vq, qx, qy, qz, px, py, pz, d1);
        if(d1<d_1) imax=imax-1;
        v_1=c+double(jmax-1)*vh; u0=a+double(imax)*uh;
        GetQSurDerivat(0, 0, u0, v_1, qx, qy, qz);
        uq=u0; vq=v_1;
        QtoPSurDistance(uq, vq, qx, qy, qz, px, py, pz, d_1);
        v1=c+double(jmax+1)*vh; u0=a+double(imax)*uh;
        GetQSurDerivat(0, 0, u0, v1, qx, qy, qz);
        uq=u0; vq=v1;
        QtoPSurDistance(uq, vq, qx, qy, qz, px, py, pz, d1);
        if(d1<d_1) jmax=jmax-1;
    }
    if((jmax>0&&jmax<10&&imax==0) || (jmax>0&&jmax<10&&imax==10))
    {
        if(imax==0) u0=a;
        if(imax==10) u0=a+double(imax)*uh;
        v_1=c+double(jmax-1)*vh; v1=c+double(jmax+1)*vh;
        GetQSurDerivat(0, 0, u0, v_1, qx, qy, qz);
        uq=u0; vq=v_1;
        QtoPSurDistance(uq, vq, qx, qy, qz, px, py, pz, d_1);
        uq=u0; vq=v1;
    }
}
```

```

SurQtoPHausdorff. txt
QtoPSurDistance(uq, vq, qx, qy, qz, px, py, pz, d1);
if(d1<d_1) jmax=jmax-1;
if(imax==10) imax=9;
}
if((imax>0&&imax<10&&jmax==0) || (imax>0&&imax<10&&jmax==10))
{
    if(jmax==0) v0=c;
    if(jmax==10) v0=c+double(jmax)*vh;
    u_1=a+double(imax-1)*uh; u1=a+double(imax+1)*uh;
    GetQSurDerivat(0, 0, u_1, v0, qx, qy, qz);
    uq=u_1; vq=v0;
    QtoPSurDistance(uq, vq, qx, qy, qz, px, py, pz, d_1);
    uq=u1; vq=v0;
    QtoPSurDistance(uq, vq, qx, qy, qz, px, py, pz, d1);
    if(d1<d_1) imax=imax-1;
    if(jmax==10) jmax=9;
}
/* 以下计算最大偏差所在子矩形域四角点的偏差d00, d01, d0, d11. */
uq=a+double(imax)*uh; vq=c+double(jmax)*vh;
QtoPSurDistance(uq, vq, qx, qy, qz, px, py, pz, d00);
uq=a+double(imax+1)*uh; vq=c+double(jmax)*vh;
QtoPSurDistance(uq, vq, qx, qy, qz, px, py, pz, d10);
uq=a+double(imax)*uh; vq=c+double(jmax+1)*vh;
QtoPSurDistance(uq, vq, qx, qy, qz, px, py, pz, d01);
uq=a+double(imax+1)*uh; vq=c+double(jmax+1)*vh;
QtoPSurDistance(uq, vq, qx, qy, qz, px, py, pz, d11);
/* 在下面采用黄金分割法的while循环对最大距离所在子矩形域进行最大距离的精搜索, nn为执行的迭代次数
*/
u0=a+imax*uh; v0=c+jmax*vh; u1=a+(imax+1)*uh; v1=c+(jmax+1)*vh;
int nn=0;
while(u0>a+imax*uh&&u1<=a+(imax+1)*uh&&v0>=c+jmax*vh&&v1<=
    c+(jmax+1)*vh&&fabs(u1-u0)>eps&&fabs(v1-v0)>eps)
{
    nn=nn+1;
    if(d00+d01>=d10+d11&&d00+d10>=d01+d11)
    {
        u1=u0+ratio*(u1-u0); v1=v0+ratio*(v1-v0);
        GetQSurDerivat(0, 0, u1, v1, qx, qy, qz);
        up1=u1; vp1=v1;
        QtoPSurDistance(up1, vp1, qx, qy, qz, px, py, pz, d11);
        if(d11>maxd) {maxd=d11; maxqx=qx; maxqy=qy; maxqz=qz;
            maxpx=px; maxpy=py; maxpz=pz;}
    }
    if(d00+d01<d10+d11&&d00+d10<d01+d11)
    {
        u0=u0+(1-ratio)*(u1-u0); v0=v0+(1-ratio)*(v1-v0);
        GetQSurDerivat(0, 0, u0, v0, qx, qy, qz);
        up0=u0; vp0=v0;
        QtoPSurDistance(up0, vp0, qx, qy, qz, px, py, pz, d00);
        if(d00>maxd) {maxd=d00; maxqx=qx; maxqy=qy; maxqz=qz;
            maxpx=px; maxpy=py; maxpz=pz;}
    }
    if(d00+d01>=d10+d11&&d00+d10<d01+d11)
    {
        u1=u0+ratio*(u1-u0); v0=v0+(1-ratio)*(v1-v0);
        GetQSurDerivat(0, 0, u1, v0, qx, qy, qz);
        up1=u1; vp0=v0;
        QtoPSurDistance(up1, vp0, qx, qy, qz, px, py, pz, d10);
        if(d10>maxd) {maxd=d10; maxqx=qx; maxqy=qy; maxqz=qz;
            maxpx=px; maxpy=py; maxpz=pz;}
    }
    if(d00+d01<d10+d11&&d00+d10>=d01+d11)
    {
        u0=u0+(1-ratio)*(u1-u0); v1=v0+ratio*(v1-v0);
        GetQSurDerivat(0, 0, u0, v1, qx, qy, qz);
        up1=u0; vp1=v1;
        QtoPSurDistance(up1, vp1, qx, qy, qz, px, py, pz, d01);
        if(d01>maxd) {maxd=d01; maxqx=qx; maxqy=qy; maxqz=qz;
            maxpx=px; maxpy=py; maxpz=pz;}
    }
}
}

```